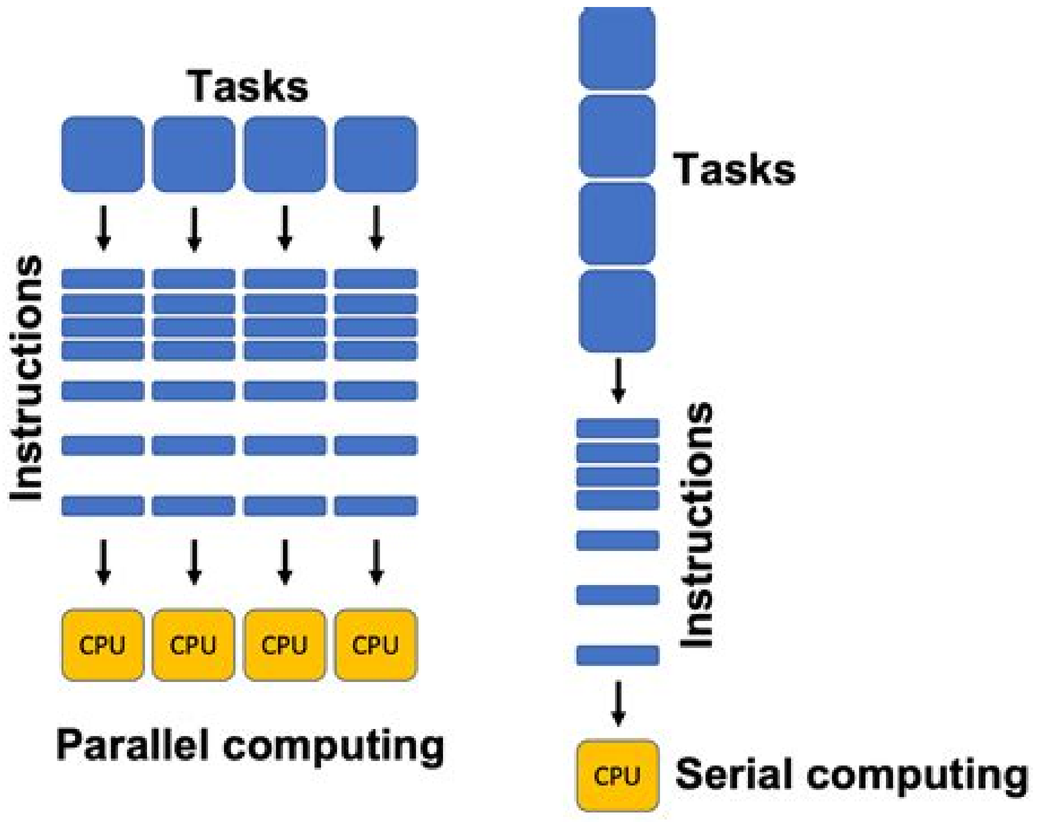
Exercises UD01



1. Exercises

1. Create a project called `ProcessListPNG` with a program that asks the user to introduce a path (for instance, `/myfolder/photos`), and then launches a process that prints a list of all PNG images found in this path. Try to do it recursively (either with a command from the operating system or with your own script).
2. Create a Java program that uses the `waitFor()` method to check the output value of the running process. It tests the execution of the program Notepad (or similar) two times. First time the user will close the windows normally so Notepad return 0 and our program says OK. And the second time the user will kill Notepad from Operating System Task manager and out program would say ERROR.

```
OK(0)
ERROR(1)
```

~~Create a Java program called ReadName.java that receives a name from the main() arguments and displays it on the screen. It uses System.exit(1) for a correct termination of the program and System.exit(-1) in case the correct arguments were not entered in main().~~

~~Next, create a program similar to the following:~~

```java
package UD01;
import java.io.*;

 /**
  *
  * @author David Martínez (wwww.martinezpenya.es|iesmre.com)
  */
 public class Test {

     public static void main(String[] args) throws IOException {
         //create File Object where the example is located
         File folder = new File(".\\bin");
         //process to execute
         ProcessBuilder pb = new ProcessBuilder("java", "example");
         //move to that folder
         pb.directory(folder);
         System.out.format("Working folder: %s%n", pb.directory());
         //run the process
         Process p = pb.start();
         //get the output of the process
         try {
             InputStream is = p.getInputStream();
             int c;
             while ((c = is.read()) != -1) {
                 System.out.println((char) c);
             }
             is.close();
```

```java
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

~~to execute ReadName.java. It uses the `waitFor()` method to check the output value of the running process. It tests the execution of the program giving value to the arguments of `main()` and without giving it value. What does `waitFor()` return in one case and in another?~~

3. Create a project called `ProcessKillNotepad` with a program that launches the notepad or any similar text editor from your operating system. Then, the program will wait 10 seconds for the subprocess to finish and, after that period, it will be destroyed. To sleep 10 seconds, use this instruction:

```java
Thread.sleep(milliseconds);
```

4. Create a project called `DishWasher`. We are going to simulate a dish washing process at home, when someone wash the dishes and someone else dries them. Create the following classes:
   - A `Dish` class with just an integer attribute: the dish number (to identify the different dishes).
   - A `DishPile` class that will store up to 5 dishes. It will have a wash method that will put a dish in the pile (if there is space available), and a dry method that will take a dish from the pile (if there is any). Maybe you will need a Dish parameter in wash method, to add a dish to the pile.
   - A `Washer` thread that will receive a number N as a parameter, and a `DishPile` object. In its run method, it will put (wash) N dishes in the pile, with a pause of 50ms between each dish.
   - A `Dryer` thread that will receive a number N as a parameter, and a `DishPile` object. In its run method, it will take out (dry) N dishes from the pile, with a pause of 100 ms between each dish.
   - The `main` class will create the `DishPile` object, and a thread of each type (`Washer` and `Dryer`). They will have to wash/dry 20 dishes coordinately, so that the output must be something like this:

```
Washed dish #1, total in pile: 1
Drying dish #1, total in pile: 0
Washed dish #2, total in pile: 1
Drying dish #2, total in pile: 0
Washed dish #3, total in pile: 1
Washed dish #4, total in pile: 2
Drying dish #4, total in pile: 1
Washed dish #5, total in pile: 2
Washed dish #6, total in pile: 3
Drying dish #6, total in pile: 2
Washed dish #7, total in pile: 3
...
```

2. Activities

1. The objective of this activity is to understand the problem that concurrent programming poses when protection methods are not applied on critical sections.

   We have a system that controls the coins that enter the parking meters every hour to control the time of day with the most traffic in the parking lot.

   To do this the parking meters have two processes running in their system:

   - A counter process, which will count the coins that enter the parking meter.
   - Another process that every hour sends the number of coins to the central and sets the number of coins to zero.

   There is a shared variable that will have the number of coins in the parking meter. They are two processes that run concurrently, so it is possible that at a given moment they run at the same time.

   It indicates the possible orders of execution of instructions 1, 2 and 3 of the two processes if they are executed concurrently. Also indicate which of the possible orders would be correct.

   1. `sendHeadOffice(currency)`
   2. `currency=0`
   3. `currency=currency+1`

   The code for the processes would be as follows:

```
void enviaNumeroModenes(){
  while (true){
    enviarOficinaCentral(moneda)
    moneda=0
    }
       }
   void comptadorModenes(){
    while (entradaModena==true)
        moneda=moneda+1
     }
   }
 }
```

2. Say whether the following statements are true (T) or false (F):

No	Question	V	F
1	A process always has a state		
2	A process and an application are synonymous		
3	In concurrent programming a processor can execute different processes at the time		

3. Answer the following questions:

- Write some characteristic of a concurrent program.
    - What is the advantage of concurrency in uniprocessor systems?
    - What are the differences between multiprogramming and multithreading?
    - What are the two main problems inherent in concurrent programming?

4. Go to the following URL https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial and answer the following questions:
    - Mention some characteristics of serial computing.
    - Mention some characteristics of parallel computing.
    - Areas in which parallel computing is used.
    - How does the SETI@home project make use of parallel computing?

5. Which of the following operating systems is not multitasking?

   a) Unix.

   b) Linux.

   c) Windows 10.

   d) MS-DOS.

6. In computing, multitasking is understood as:

   a) The ability of a computer to perform complex tasks.

   b) The ability of a computer to run one program after another.

   c) The ability of a computer to perform tasks synchronized with each other.

   d) The ability of a computer to run several programs at the same time.

7. What element of the computer is responsible for managing multitasking?

   a) The programming language chosen to create the programs.

   b) The operating system.

   c) The file system.

   d) None of the above answers is correct.

8. Which statement regarding compiled programming languages is not correct?

   a) Under normal conditions they are faster than the interpreted ones.

   b) They require a specific compilation for each operating system.

   c) If the language is standard, the same compilation works for all operating systems.
   d) The source code must be compiled to obtain the binary code to be executed.

9. The process identifier is usually identified by the acronyms:

   a) CPU.

   b) PID.

   c) IPD.

   d) TIY.

10. Processing running on different independent but connected computers
    and synchronized is called:

a) Distributed.

b) Concurrent.

c) Parallel.

d) Monohilo.

11. Which of the following is not a goal of the operating system process scheduler?

    a) Maximize system performance.

    b) Maximize response times.

    c) Maximize equity in the distribution of resources.

    d) Minimize waiting times.

12. What is the state to which a process that is in the Ready state can go?

    a) New.

    b) Blocked.

    c) Running.

    d) Finished.

13. Which statement regarding the Java Runtime class is wrong?

    a) It allows launching a process indicating the execution parameters.

    b) Allows to wait for the termination of the launched process.

    c) It allows knowing the value of the internal variables of the launched process.

    d) It allows obtaining the status of the completion of the launched process.

14. How do you call the method of the Java Process class that makes you wait for the process launched?

    a) `sleep()`.

    b) `finish()`.

    c) `waitFor()`.

    d) `waitEnd()`.

3. Application activities

1. Explain the differences between interpreted and compiled programming languages.
2. Describe the differences between program and process.
3. Explain what distributed programming consists of.
4. Find the differences between running two processes or performing a fork.
5. Design a database-based system to exchange information between two processes.
6. Design a system based on text files to synchronize processes.
7. Decide which synchronization system could be used to synchronize processes running on stand-alone computers connected via the internet.
8. Explain why threads consume less resources than processes.
9. Explain what task the process scheduler performs within the operating system.
10. Design a simple process planner. It determines how CPU time is allocated to the various processes it manages so that they can all proceed more or less simultaneously.
11. Prepare the list of values and their descriptions that can be returned by the execution of a process that has to access a database, a computer through the network and the Internet.
12. Write a Java program that runs the **Firefox** web browser. It uses the `Runtime` class. Tries the browser to directly open a web page.

4. Information sources

- Wikipedia
- Programación de servicios y procesos - FERNANDO PANIAGUA MARTÍN [Paraninfo]
- Programación de Servicios y Procesos - ALBERTO SÁNCHEZ CAMPOS [Ra-ma]
- Programación de Servicios y Procesos - Mª JESÚS RAMOS MARTÍN - [Garceta] (1ª y 2ª Edición)
- Programación de servicios y procesos - CARLOS ALBERTO CORTIJO BON [Sintesis]
- Programació de serveis i processos - JOAR ARNEDO MORENO, JOSEP CAÑELLAS BORNAS i JOSÉ ANTONIO LEO MEGÍAS [IOC]
- GitHub repositories:
  - https://github.com/ajcpro/psp
  - https://oscarmaestre.github.io/servicios/index.html
  - https://github.com/juanro49/DAM/tree/master/DAM2/PSP
  - https://github.com/pablohs1986/dam_psp2021
  - https://github.com/Perju/DAM
  - https://github.com/eldiegoch/DAM
  - https://github.com/eldiegoch/2dam-psp-public
  - https://github.com/franlu/DAM-PSP
  - https://github.com/ProgProcesosYServicios
  - https://github.com/joseluisgs
  - https://github.com/oscarnovillo/dam2_2122
  - https://github.com/PacoPortillo/DAM_PSP_Tarea02_La-Cena-de-los-Filosofos